INFINITE TIME TURING MACHINES AND AN APPLICATION TO THE HIERARCHY OF EQUIVALENCE RELATIONS ON THE REALS

SAMUEL COSKEY AND JOEL DAVID HAMKINS

ABSTRACT. We describe the basic theory of infinite time Turing machines and some recent developments, including the infinite time degree theory, infinite time complexity theory, and infinite time computable model theory. We focus particularly on the application of infinite time Turing machines to the analysis of the hierarchy of equivalence relations on the reals, in analogy with the theory arising from Borel reducibility. We define a notion of infinite time reducibility, which lifts much of the Borel theory into the class $\undertilde{\Delta}^1_2$ in a satisfying way.

Infinite time Turing machines fruitfully extend the operation of ordinary Turing machines into transfinite ordinal time and by doing so provide a robust theory of computability on the reals. In a mixture of methods and ideas from set theory, descriptive set theory and computability theory, the approach provides infinitary concepts of computability and decidability on the reals, which climb nontrivially into the descriptive set-theoretic hierarchy (at the level of Δ^1_2) while retaining a strongly computational nature. With infinite time Turing machines, we have infinitary analogues of numerous classical concepts, including the infinite time Turing degrees, infinite time complexity theory, infinite time computable model theory, and now also the infinite time analogue of the theory of Borel equivalence relations under Borel reducibility.

In this article, we shall give a brief review of the machines and their basic theory, and then explain in a bit more detail our recent application of infinite time computability to an analogue of Borel equivalence relation theory, a full account of which is given in [CH11]. The basic idea of this application is to replace the concept of Borel reducibility commonly used in that theory with forms of infinite time computable reducibility, and study the accompanying hierarchy of equivalence relations. This approach retains much of the Borel analysis and results, while also illuminating a part of the hierarchy of equivalence relations that seems beyond the reach of the Borel theory, including many highly canonical

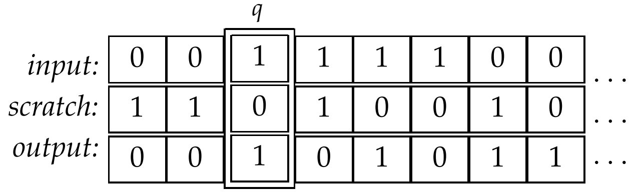
The second author's research has been supported in part by grants from the Research Foundation of CUNY and from the National Science Foundation.

equivalence relations that are infinite time computable but not Borel, such as the isomorphism relations for diverse classes of countable structures.

Major parts of this article are adapted from the surveys [Ham07] and [Ham05] and from our article [CH11] on infinite time computable equivalence relations. Infinite time Turing machines were first studied by Hamkins and Kidder in 1989, with the core introduction provided by Hamkins and Lewis [HL00]. The theory has now been extended by many others, including Philip Welch, Peter Koepke, Benedikt Löwe, Daniel Seabold, Ralf Schindler, Vinay Deolalikar, Russell Miller, Steve Warner, Giacomo Lenzi, Erich Monteleone, Samuel Coskey and others. Numerous precursors to the theory include Blum-Shub-Smale machines (1980s), Büchi machines (1960s) and accompanying developments, Barry Burd's model of Turing machines with "blurs" at limits (1970s), the extensive development of α-recursion and E-recursion theory, a part of higher recursion theory (since the 1970s), Jack Copeland's accelerated Turing machines (1990s), Ryan Bissell-Siders' ordinal machines (1990s), and more recently, Peter Koepke's ordinal Turing machines and ordinal register machines (2000s). The expanding literature involving infinite time Turing machines includes [HL00], [Wel99], [Wel00a], [Wel00b], [LÖ1], [HS01], [HL02], [Sch03], [HW03], [Ham02], [Ham04], [LM04], [DHS05], [HMSW07], [Ham05], [Wel], [Wel05], [Koe05], [Ham07], [HM09], [HM07], [HLM07] and others.

1. A Brief Review of Infinite time Turing machines

Infinite time Turing machines have exactly the same hardware as their classical finite time counterparts, with a head moving back and forth on a semi-infinite paper tape, writing 0s and 1s according to the rigid instructions of a finite program with finitely many states. What is new about the infinite time Turing machines is that their operation is extended into transfinite ordinal time. For convenience, the machines are implemented with a three-tape model, with separate tapes for input, scratch work and output. The machine


q
input: 0 0 1 1 1 1 0 0 ...
scratch: 1 1 0 1 0 0 1 0 ...
output: 0 0 1 0 1 0 1 1 ...

operates at successor stages of computation in exactly the classical manner, according to the program instructions. Computation is extended to limit ordinal stages simply by defining the limit configuration of the machines. The idea is to try to preserve as much as possible the information that the computation has been creating up to that stage, preserving it in the limit configuration as a kind of limit of the earlier configurations. Specifically, at any limit ordinal stage ξ, the machine enters what we call the *limit* state, one of the distinguished states along with the *start* and *halt* states; the head is reset to the first cell at the left; and each cell of the tape is updated with the $\limsup$ of the values previously displayed in that cell. Having thus specified the complete configuration of the machine at stage ξ, the computation may now continue to stage $\xi + 1$ and so on. Computational output is given only when the machine explicitly enters the *halt* state, and computation ceases when this occurs.

Since the tapes naturally accommodate infinite binary strings—and there is plenty of time for the head to inspect every cell—the natural context for input and output to the machines is Cantor space 2^ω, which we denote by $\mathbb{R}$ and refer to as the reals. Thus, the machines provide an infinitary notion of computability on the reals. A program p computes the partial function $\varphi_p \colon \mathbb{R} \to \mathbb{R}$, defined by $\varphi_p(x) = y$ if program p on input x yields output y, where the output of a computation is the content of the output tape when the machine enters the *halt* state. A subset $A \subseteq \mathbb{R}$ is *infinite time decidable* if the characteristic function of A is infinite time computable. The set A is *infinite time semi-decidable* if the constant partial function $1 \upharpoonright A$ is computable. This is equivalent to A being the domain of an infinite time computable function (but not necessarily equivalent to A being the range of such a function). Elementary results in [HL00] show that the arithmetic sets are exactly those that are decidable in time uniformly less than ω^2 and the hyperarithmetic sets are those that are decidable in time less than some recursive ordinal. The power of the machines, however, reaches much higher than this into the descriptive set theoretic hierarchy.

For example, every Π^1_1 and Σ^1_1 set is infinite time decidable. To see this, it suffices to show that the complete Π^1_1 set WO, consisting of reals coding a well-ordered relation on ω, is infinite time computable. This is accomplished by the *count-through* argument of [HL00, Theorem 2.2], which we should like to sketch here. Given a real x, we view it as coding the relation $\lhd$ on ω for which $n \lhd m$ if and only if the $\langle n, m\rangle$ bit of x is 1. The assertion that $\lhd$ is a linear order is arithmetic in x, and therefore easily determined by the machines. After this, the machine will check for well-foundedness essentially by counting through the order, relying on the fact that the computational steps are themselves well-ordered. Specifically, the machine places an initial guess for the current minimal element in the relation $\lhd$, updating it with better guesses as they are encountered. At each revision, the machine flashes a certain master flag, so that at the limit stage the machine can know if the guess was changed infinitely often, indicating ill-foundedness (the machine should reset the master flag at limits of limit stages). Otherwise, the true current minimal element has been found, and so the machine can delete all mention of it from the field of the relation coded by x. Iterating this, the algorithm in effect systematically erases the well-founded initial segment of the relation coded by the input real, until either nothing is left or the ill-founded part is discovered, either of which can be determined. In this way, membership in WO is infinite time decidable. It follows that every Π^1_1 and Σ^1_1 set is infinite time decidable, and so the machines climb properly into Δ^1_2. Meanwhile, the class of infinite time decidable sets is easily observed to be contained in Δ^1_2, and in fact the class Δ^1_2 is closed under the infinite time jump operations and is therefore stratified by a significant part of the infinite time Turing degrees.

Although transfinite, computations are nevertheless inherently countable, since an easy cofinality argument establishes that every computation either halts or repeats by some countable ordinal stage. An ordinal α is said to be *clockable*, if there is a computation $\varphi_p(0)$ halting on exactly the α^{th} step. A real x is *writable* if it is the output of a computation $\varphi_p(0)$, and an ordinal is writable if it is coded by such a real. Because there are only countably many programs, it follows that there are only countably many clockable and writable ordinals. The clockable and writable ordinals extend through all the recursive ordinals and far beyond; their supremum is recursively inaccessible and more. The writable ordinals form an initial segment of the ordinals, since whenever an ordinal is writable, then the algorithm writing it can be easily modified to write a code for any smaller ordinal. But the same is not true for the clockable ordinals; in the midst of the clockable ordinals, there are increasingly complex forbidden regions at which no (parameter-free) infinite time Turing machine can halt.

Let us quickly sketch the argument that such gaps in the clockable ordinals exist, since this is an interesting exercise in ordinal reflection that constitutes a basic method of many later constructions in the theory. Consider the algorithm that simulates all programs on input 0 simultaneously, by some bookkeeping method that reserves and manages sufficient separate space for each, simulating ω many steps of computation for each program in each ω many steps of actual computation. Our algorithm might keep careful track of which programs have halted, and pay attention to find a stage at which none of the programs halt. Since such a stage exists above the supremum of all clockable ordinals, we will definitely find such a stage eventually. Since our algorithm can recognize the first such stage, we can arrange that it halts immediately after this discovery. So we have described a computational procedure that will halt at an ordinal stage that is larger than a stage at which no computations halted, and so there are gaps in the clockable ordinals, as desired. A careful analysis of the algorithm shows that the first gap after any clockable ordinal has order type ω, essentially because it takes ω many additional steps to realize that a gap has been reached. Modified algorithms search for longer gaps and show that there must be increasingly complex gaps at increasingly complex admissible limit stages—for any clockable or writable ordinal α, there are gaps of size at least α. The structure of these gaps exhibits the same complexity as the infinite time halting problem.

Although it was established in [HL00] that the clockable and writable ordinals have the same order type, perhaps the main question left open in that paper was whether the supremum of these ordinals was the same. This was settled in the affirmative by Philip Welch in [Wel00b]. Another way to describe the result is that whenever program p on input x yields a halting computation, then there is another computation that writes out a certificate of this computation, a real coding the entire computation history including a well-ordered relation whose order type is the length of the computation. This important fact, far from obvious, relies on a subtle treatment of eventual writability and constitutes a foundation of many further developments of the theory, including the applications we mention in this article.

The reflective aspect of the count-through argument described above consists of the observation that any decidable property that holds of a real that might be encountered during the course of a computation must hold of a writable real, since we may embark on the computational search to find such a witness and output it when it is found. This idea is greatly extended by the λ-ζ-Σ theorem of Philip Welch. Specifically, [HL00] defines that a real x is *eventually writable* if there is a computation $\varphi_p(0)$ for which x appears on the output tape from some point on (even if the computation does not halt), and x is *accidentally writable* if it appears on any of the tapes at any stage during a computation $\varphi_p(0)$. By coding ordinals with reals, we obtain the notions of eventually and accidentally writable ordinals. If λ is the supremum of the clockable or writable ordinals, ζ is the supremum of the eventually writable ordinals and Σ is the supremum of the accidentally writable ordinals, then [HL00] establishes $\lambda < \zeta < \Sigma$. The λ-ζ-Σ theorem of Welch [Wel00a] asserts moreover that $L_\lambda \prec_{\Sigma_1} L_\zeta \prec_{\Sigma_2} L_\Sigma$, using the initial segments of Gödel's constructible universe, and furthermore, that these ordinals are characterized as the least example of this pattern. This result precisely expresses the sense in which the algorithms may pull down witnesses from the accidentally writable realm into the eventually writable or writable realms. At the heart of the proof and the result is the fact that every computation repeats the stage ζ configuration at stage Σ.

Many of the fundamental constructions of classical finite time computability theory carry over to the infinite time context. For example, one can prove the infinite time analogues of the *smn*-theorem, the Recursion theorem and the undecidability of the infinite time halting problem, by essentially the classical arguments. Some other classical facts, however, do not directly generalize. For example, it is not true in the infinite time context that if the graph of a function f is semi-decidable, then the function is computable. This is a consequence of the following:

Theorem 1 (Lost Melody Theorem)**.** *There is a real c such that $\{c\}$ is infinite time decidable, but c is not writable.*

The real c, a lost melody that you cannot sing on your own, although you can recognize it yes-or-no when someone sings it to you, exhibits sufficient internal structure that $\{c\}$ is decidable, but is too complicated itself to be writable. That is, we can recognize whether a given real y is c or not, but we cannot produce c from nothing. The function $f(x) = c$ with constant value c, therefore, is not computable, because c is not writable, but the graph is decidable, because we can recognize whether a pair has the form (x, c).

The infinite time analogue of the halting problem breaks into lightface and boldface versions, $h = \{\, p \mid \varphi_p(p)\downarrow \,\}$ and $H = \{\, (p, x) \mid \varphi_p(x)\downarrow \,\}$, respectively. These are both semi-decidable and not decidable, but in the infinitary context, they are not computably equivalent.

The notion of oracle computation lifts to the infinitary context and gives rise to a theory of relative computability and a rich structure of degrees. In contrast to the classical theory on $\mathbb{N}$, however, in the infinite time context we have two natural sorts of oracles to be used in oracle computations, corresponding to the second order nature of the theory. First, one can use an individual real as an oracle in exactly the classical manner, by adjoining an oracle tape on which the values of that real are written out. This amounts to fixing a supplemental input parameter and can be viewed as giving rise to a boldface theory of infinitary computability, just as one allows arbitrary real parameters in the descriptive set-theoretic treatment of boldface $\utilde{\Delta}^1_1$ and $\utilde{\Pi}^1_1$. (We shall explicitly adopt such a boldface perspective in our application to the theory of equivalence relations under infinite time reducibility.) Second, however, one naturally wants somehow to use a *set* of reals as an oracle, although we cannot expect in general to write such a set out on the tape (perhaps it is even uncountable). Instead, the oracle tape is empty at the start of computation, and during the computation the machine may freely write on this tape; whenever the algorithm calls for it, the machine may make a membership query about whether the real currently written on the oracle tape is a member of the oracle or not. Thus, the machine is able to know of any real that it can produce, whether the real is in the oracle set or not.

Such oracle computations give rise to a notion of relative computability $\varphi_p^A(x)$ and therefore a notion of infinite time omputable reduction $A \leq_\infty B$ and the accompanying infinite time degree relation $A \equiv_\infty B$. For any set A, we have the lightface jump $A^\triangledown$ and the boldface jump $A^\blacktriangledown$, corresponding to the two halting problems, relativized to A. The boldface jump jumps much higher than the lightface jump, as [HL00] establishes that $A <_\infty A^\triangledown <_\infty A^\blacktriangledown$, as well as $A^{\triangledown\blacktriangledown} \equiv_\infty A^\blacktriangledown$ and a great number of other interesting interactions. The infinite time analogue of Post's problem, the question of whether there are intermediate semi-decidable degrees between 0 and the jump $0^\triangledown$, was settled by [HL02] in an answer that cuts both ways:

Theorem 2. *The infinite time analogue of Post's problem has both affirmative and negative solutions.*

1. *There are no reals z with $0 <_\infty z <_\infty 0^\triangledown$.*
2. *There are sets of reals A with $0 <_\infty A <_\infty 0^\triangledown$. Indeed, there are incomparable semi-decidable sets of reals $A \perp_\infty B$.*

The degrees of the accidentally writable reals are linearly ordered and in fact form a well-ordered hierarchy of order type $\zeta + 1$, which corresponds also to their order of earliest appearance on any computation. In other work, Welch [Wel99] found minimality in the infinite time Turing degrees. Hamkins and Seabold [HS01] analyzed one-tape versus multi-tape infinite time Turing machines, and Benedikt Löwe [LÖ1] observed the connection between infinite time Turing machines and revision theories of truth.

2. Some applications and extensions

Let us briefly describe a few of the recent developments and extensions of infinite time Turing machines, such as the rise of infinite time complexity theory and the introduction of infinite time computable model theory. After this, in the following section we shall go into greater detail concerning the application of infinite time Turing machines to an analogue of the theory of Borel equivalence relations.

Ralf Schindler [Sch03] initiated the study of infinite time complexity theory by solving the infinite time Turing machine analogue of the P versus NP question. To define the polynomial class P in the infinite time context, Schindler observed simply that all reals have length ω and the polynomial functions of ω are bounded by those of the form ω^n. Thus, he defined that a set $A \subseteq \mathbb{R}$ is in P if there is a program p and a natural number n

such that p decides A and halts on all inputs in time before ω^n. The set A is in NP if there is a program p and a natural number n such that $x \in A$ if and only if there is y such that p accepts (x, y), and p halts on all inputs in time less than ω^n. Schindler proved $\mathrm{P} \neq \mathrm{NP}$ for infinite time Turing machines in [Sch03], using methods from descriptive set theory to analyze the complexity of the classes P and NP. This has now been generalized in joint work [DHS05] to the following, where the class co-NP consists of the complements of sets in NP.

Theorem 3. $\mathrm{P} \neq \mathrm{NP} \cap \text{co-NP}$ *for infinite time Turing machines.*

This proof appears in [DHS05]. It follows that $\mathrm{P} \neq \mathrm{NP}$ for infinite time Turing machines. (This result has no bearing whatsoever on the finitary classical $\mathrm{P} \neq \mathrm{NP}$ question.) Some of the structural reasons behind $\mathrm{P} \neq \mathrm{NP} \cap \text{co-NP}$ are revealed by placing the classes P and NP within a larger hierarchy of complexity classes P_α and NP_α using computations of size bounded below α. Results in [DHS05] showed, for example, that the classes NP_α are identical for $\omega + 2 \leq \alpha \leq \omega_1^{\mathrm{CK}}$, but nevertheless, $\mathrm{P}_{\alpha+1} \subsetneq P_{\alpha+2}$ for any clockable limit ordinal α. It follows, since the P_α are steadily increasing while the classes $\mathrm{NP}_\alpha \cap \text{co-NP}_\alpha$ remain the same, that $\mathrm{P}_\alpha \subsetneq \mathrm{NP}_\alpha \cap \text{co-NP}_\alpha$ for any ordinal α with $\omega + 2 \leq \alpha < \omega_1^{\mathrm{CK}}$. Thus, $\mathrm{P} \neq \mathrm{NP} \cap \text{co-NP}$. Nevertheless, we attain equality at the supremum ω_1^{CK} with

$$\mathrm{P}_{\omega_1^{\mathrm{CK}}} = \mathrm{NP}_{\omega_1^{\mathrm{CK}}} \cap \text{co-NP}_{\omega_1^{\mathrm{CK}}}.$$

In fact, this is an instance of the equality $\Delta_1^1 = \Sigma_1^1 \cap \Pi_1^1$, and one can thereby begin to see how the theory of infinite time Turing machines grows naturally into descriptive set theory.

This same pattern of inequality $\mathrm{P}_\alpha \subsetneq \mathrm{NP}_\alpha \cap \text{co-NP}_\alpha$ is mirrored higher in the hierarchy, whenever α lies strictly within a contiguous block of clockable ordinals, with the corresponding $\mathrm{P}_\beta = \mathrm{NP}_\beta \cap \text{co-NP}_\beta$ for any β that begins a gap in the clockable ordinals. In addition, the question is settled in [DHS05] for the other complexity classes P^+, P^{++} and P^f. Benedikt Löwe has introduced analogues of PSPACE.

The subject of infinite time computable model theory was introduced in [HMSW07]. Computable model theory is model theory with a view to the computability of the structures and theories that arise. Infinite time computable model theory carries out this program with the notion of infinite time computability provided by infinite time Turing machines. The classical theory began decades ago with such topics as computable completeness (Does every decidable theory have a decidable model?) and computable categoricity (Does every isomorphic pair of computable models have a computable isomorphism?), and the field has now matured into a sophisticated analysis of the complexity spectrum of countable models and theories.

The motivation for a broader context is that, while classical computable model theory is necessarily limited to countable models and theories, the infinitary computability context allows for uncountable models and theories, built on the reals. Many of the computational constructions in computable model theory generalize from structures built on $\mathbb{N}$, using finite time computability, to structures built on $\mathbb{R}$, using infinite time computability. The uncountable context opens up new questions, such as the infinitary computable Löwenheim-Skolem Theorem, which have no finite time analogue. Several of the most natural questions turn out to be independent of ZFC.

In joint work [HMSW07], we defined that a model $\mathcal{A} = \langle A, \ldots \rangle$ is infinite time *computable* if $A \subseteq \mathbb{R}$ is decidable and all functions, relations and constants are uniformly infinite time computable from their Gödel codes and input. The structure $\mathcal{A}$ is *decidable* if one can compute whether $\mathcal{A} \models \varphi[\bar{a}]$ given $\ulcorner\varphi\urcorner$ and $\bar{a}$. A theory T is infinite time *decidable* if the relation $T \vdash \varphi$ is computable in $\ulcorner\varphi\urcorner$. Because we want to treat uncountable languages, the natural context for Gödel codes is $\mathbb{R}$ rather than $\mathbb{N}$.

The initial question, of course, is the infinite time computable analogue of the Completeness Theorem: Does every consistent decidable theory have a decidable model? The answer turns out to be independent of ZFC.

Theorem 4 ([HMSW07]). *The infinite time computable analogue of the Completeness Theorem is independent of* ZFC. *Specifically:*

(1) *If $V = L$, then every consistent infinite time decidable theory has an infinite time decidable model, in a computable translation of the language.*
(2) *It is relatively consistent with* ZFC *that there is an infinite time decidable theory, in a computably presented language, having no infinite time computable or decidable model in any translation of the language.*

The proof of (1) uses the concept of a *well-presented* language $\mathcal{L}$, for which there is an enumeration of the symbols $\langle s_\alpha \mid \alpha < \delta \rangle$ such that from any $\ulcorner s_\alpha \urcorner$ one can uniformly compute a code for the prior symbols $\langle \ulcorner s_\beta \urcorner \mid \beta \leq \alpha \rangle$. One can show that every consistent decidable theory in a well-presented language has a decidable model, and if $V = L$, then every computable language has a well presented computable translation. For (2), one uses the theory T extending the atomic diagram of $\langle \mathrm{WO}, \equiv \rangle$ while asserting that f is a choice function on the $\equiv$ classes. This is a decidable theory, but for any computable model

$\mathcal{A} = \langle A, \equiv, f \rangle$ of T, the set $\{\, f(c_u) \mid u \in \mathrm{WO} \,\}$ is Σ^1_2 and has cardinality ω_1. It is known to be consistent with ZFC that no Σ^1_2 set has size ω_1.

For the infinite time analogues of the Löwenheim-Skolem Theorem, we proved for the upward version that every well presented infinite time decidable model has a proper elementary extension with a decidable presentation, and for the downward version, every well presented uncountable decidable model has a countable decidable elementary substructure. There are strong counterexamples to a full direct generalization of the Löwenheim-Skolem theorem, however, because [HMSW07] provides a computable structure $\langle \mathbb{R}, U \rangle$ on the entire set of reals, which has no proper computable elementary substructure.

Some of the most interesting work involves computable quotients. A structure has an infinite time computable *presentation* if it is isomorphic to a computable structure, and has a computable *quotient presentation* if it is isomorphic to the quotient of a computable structure by a computable equivalence relation (a congruence). For structures on $\mathbb{N}$, in either the finite or infinite time context, these notions are equivalent, because one can computably find the least element of any equivalence class. For structures on $\mathbb{R}$, however, computing such distinguished elements of every equivalence class is not always possible.

Question 5. *Does every structure with an infinite time computable quotient presentation have an infinite time computable presentation?*

In the finite time theory, or for structures on $\mathbb{N}$, the answer of course is Yes. But in the full infinite time context for structures on $\mathbb{R}$, the answer depends on the set theoretic background.

Theorem 6. *The answer to Question 5 is independent of* ZFC. *Specifically,*

(1) *It is relatively consistent with* ZFC *that every structure with an infinite time computable quotient presentation has an infinite time computable presentation.*
(2) *It is relatively consistent with* ZFC *that there is a structure having an infinite time computable quotient presentation, but no infinite time computable presentation.*

Let us briefly sketch some of the ideas appearing in the proof. In order to construct an infinite time computable presentation of a structure, given a computable quotient presentation, we'd like somehow to select a representative from each equivalence class, in a computably effective manner, and build a structure on these representatives. Under the set theoretic assumption $V = L$, we can attach to the L-least member of each equivalence class an escort real that is powerful enough to reveal that it is the L-least member of its class, and build a computable presentation out of these escorted pairs of reals. (In particular, the new presentation is not built out of mere representatives from the original class, since these reals may be too weak; they need the help of their escorts.) Thus, if $V = L$, then every structure with a computable quotient presentation has a computable presentation. On the other side of the independence, we prove statement 2 by the method of forcing. The structure $\langle\omega_1, <\rangle$ always has a computable quotient presentation built from reals coding well orders, but there are forcing extensions in which no infinite time computable set has size ω_1, on descriptive set theoretic grounds. In these extensions, therefore, $\langle\omega_1, <\rangle$ has a computable quotient presentation, but no computable presentation.

Let us also briefly discuss some of the alternative models of ordinal computation to which infinite time Turing machines have given rise. Peter Koepke [Koe05] introduced the *Ordinal Turing Machines*, which generalize the infinite time Turing machines by extending the tape to transfinite ordinal length. The limit rules are accordingly adjusted so that the machine can make use of this extra space. Specifically, rather than using a special *limit* state, the ordinal Turing machines simply have a fixed order on their (finitely many) states, and at any limit stage, the state is defined to be the lim inf of the prior states. The head position is then defined to be the lim inf of the head positions when the machine was previously in that resulting limit state. For uniformity, then, Koepke defines that the cells of the tape use the lim inf of the prior cell values (rather than lim sup as with the infinite time Turing machines). If the head moves left from a cell at a limit position, then it appears all the way to the left on the first cell.

These machines therefore provide a model of computation for functions on the ordinals, and notions of decidability for classes of ordinals. The main theorem is that the power of these machines is essentially the same as that of Gödel's constructible universe.

Theorem 7 (Koepke)**.** *The sets of ordinals that are ordinal Turing machine decidable, with finitely many ordinal parameters, are exactly the sets of ordinals in Gödel's constructible universe L.*

Several other infinitary models of ordinal computation are based on a concept of ordinal registers, and have given rise to a rich theory. See [Koe05], [KS06], [KK06], [KS09], [CFK$^+$10], [HM07], [HM09], and [HLM07].

3. Infinite time computable equivalence relation theory

Recently, we have introduced the natural analogue of Borel equivalence relation theory in which infinite time decidable relations are compared with respect to infinite time computable reduction functions. This is motivated in part by the occasional need in the study of Borel equivalence relations to go beyond Borel. Indeed, a more powerful notion of reducibility may be able to accurately compare more complex relations. In particular, we shall be able to consider the new relations which arise out of the infinite time complexity classes.

We begin with a quick introduction to the study of Borel equivalence relations. The name of the subject is somewhat of a misnomer—in fact the principle objects of study are arbitrary equivalence relations on *standard Borel spaces*, that is, sets equipped with the Borel structure of a complete separable metric space. In applications, we think of an equivalence relation as representing a classification problem from some other area of mathematics. For instance, since any group with domain $\mathbb{N}$ is determined by its multiplication function, studying the classification problem for countable groups amounts to studying the isomorphism equivalence relation on a suitable subspace of $2^{\mathbb{N}\times\mathbb{N}\times\mathbb{N}}$. For many more examples, see Section 1.2 of [ST11].

The theory of Borel equivalence relations revolves around the following key notion of complexity. If E, F are equivalence relations on standard Borel spaces X, Y, then following [FS89] and [HK96] we say that E is *Borel reducible* to F, written $E \leq_B F$, iff there exists a Borel function $f\colon X \to Y$ such that

$$x \; E \; x' \iff f(x) \; F \; f(x') \; . \tag{1}$$

Borel reducibility measures the complexity of equivalence relations not as sets of pairs, but as *classification problems*. That is, if E is Borel reducible to F, then the classification of elements of X up to E is no harder than the classification of elements of Y up to F. The by now classical and highly successful study of Borel equivalence relations consists in part of two major endeavors. First, one wishes to map out the relationships between numerous well-understood and naturally occurring equivalence relations. Second, given a real-life classification problem one should measure its complexity by comparing it against the mapped-out benchmark relations.

Some definability condition on the reduction functions (in this case that they be Borel) is necessary. Indeed, without any such restriction reducibility would always be determined by cardinalities alone. However, there are cases of natural classifications by invariants which cannot be computed by a Borel reduction function. For instance, it is $\utilde{\Delta}^1_2$ and not Borel to compute the classical Ulm invariants for a countable torsion abelian group. One might be tempted to form a theory of $\utilde{\Delta}^1_2$ reducibility, but it turns out this notion is too generous. Indeed, as we shall see below in Theorem 11, it may lump most equivalence relations together into one trivial complexity class.

We will consider here reduction functions which are computable by an infinite time Turing machine (see [CH11] for a more complete exposition). Thus, for any two equivalence relations E, F on $\mathbb{R}$, we say that E is *infinite time computably reducible* to F, written $E \leq_c F$, if there is an infinite time computable function f (freely allowing real parameters) satisfying Equation (1). Similarly, we say that E is *eventually reducible* to F, written $E \leq_e F$, if there is an eventually computable function f satisfying Equation (1). Note here that since all uncountable standard Borel spaces are Borel isomorphic, we lose no generality by restricting ourselves to equivalence relations with domain $\mathbb{R}$.

Of course, by the remarks in Section 1 (and again emphasizing that we have allowed parameters) the infinite time computable reductions include all of the Borel reductions. Thus, our theory will extend the classical theory. Conversely, many classical proofs of *non*-reducibility $E \not\leq_B F$ rely on methods such as measure, category, or forcing. Hence, they frequently "overshoot" and show that there does not exist a reduction from E to F which is Lebesgue measurable, Baire measurable, or absolutely $\underset{\sim}{\Delta}^1_2$ (discussed below), respectively. Since the infinite time computable and eventually computable functions enjoy all three of these properties, it follows in each of these cases that $E \not\leq_c F$ and even $E \not\leq_e F$, and hence not too much is "collapsed" when we pass from the $\leq_B$ hierarchy to the $\leq_c$ and $\leq_e$ hierarchies.

The infinite time notions of reducibility are very closely related to that of absolutely $\underset{\sim}{\Delta}^1_2$ reducibility, which has been treated in the literature by Hjorth and others. Recall that a subset $A \subseteq \mathbb{R}$ is said to be *absolutely* $\underset{\sim}{\Delta}^1_2$ if it is defined by equivalent $\underset{\sim}{\Sigma}^1_2$ and $\underset{\sim}{\Pi}^1_2$ formulas which remain equivalent in every forcing extension. A function $f\colon \mathbb{R} \to \mathbb{R}$ is said to be absolutely $\underset{\sim}{\Delta}^1_2$ if its diagram $\{\, (x, n) \mid f(x) \in B_n \,\}$ is absolutely $\underset{\sim}{\Delta}^1_2$ (here, B_n runs through the basic open subsets of $\mathbb{R}$). We know of very few naturally occurring cases in which there is an absolutely $\underset{\sim}{\Delta}^1_2$ reduction between two equivalence relations but not an infinite time computable reduction. And when there is an infinite time computable reduction, one can demonstrate that this is the case by simply "coding up" an algorithm which implements the witnessing reduction function. This computational approach may be more satisfying than abstractly defining a reduction function and verifying that it is $\underset{\sim}{\Delta}^1_2$ in all forcing extensions. On the other hand, we do not have any general tools for establishing non-reducibility by infinite time computable functions beyond the already established tools mentioned above, all of which establish non-reducibility by absolutely $\underset{\sim}{\Delta}^1_2$ functions already. A brief summary of results due to Hjorth and Kanovei which establish non-reducibility for absolutely $\underset{\sim}{\Delta}^1_2$ functions can be found in Section 5 of [CH11]. Some deeper results on this notion of reducibility can be found Hjorth in Chapter 9 of [Hjo00].

For an example of "coding up" a new (non-Borel) reduction function, consider the E_{ck} relation defined by $x \; E_{ck} \; y$ if x and y compute (in the ordinary sense) the same ordinals. We will compare it against the relation $\cong_{\mathrm{WO}}$, which is just the isomorphism relation restricted to the set of codes for well-orders. These two relations are not comparable by Borel reductions; nevertheless they are closely related and this is made precise by the following result.

Theorem 8. *E_{ck} and $\cong_{\mathrm{WO}}$ are infinite time computably bireducible.*

For instance, there is an intuitive reduction from E_{ck} to $\cong_{\mathrm{WO}}$—namely, map x to a code for the supremum of the ordinals which are computable (in the ordinary sense) from x. And indeed, this intuition easily translates into a program for an infinite time Turing machine. Briefly, the program simply simulates all ordinary Turing computations, and inspects the real enumerated by each. Whenever one of these reals is seen to be code for a well-order, this code is added to a list. Finally, the program computes and outputs a code for the supremum of the ordinals in its list.

Another obvious benefit to using infinite time computable and eventually computable reductions is that they are tailor-made to handle equivalence relations which arise in the study of infinite time complexity classes. As a very simple example, consider two of the most important such equivalence relations: the infinite time degree relation $\equiv_\infty$ which was introduced in Section 1, and the (light face) jump equivalence relation defined by $x \; J \; y$ if and only if $x^\triangledown \equiv_\infty y^\triangledown$. We have the following (somewhat trivial) relationship between the two.

Theorem 9. *J is eventually reducible to $\equiv_\infty$ by the function which computes the infinite time jump of a real.*

The program which witnesses this simply simulates all infinite time programs on input x, and whenever one of them halts adds its index to a list on its output tape. Since all programs which will halt do so by stage λ, the output tape will eventually show $x^\triangledown$.

Meanwhile, the next result gives a sampling of non-reducibility results which can be obtained using the methods of Hjorth and Kanovei discussed above. Here $=$ of course denotes the equality relation on $\mathbb{R}$, and E_0 the almost equality relation defined by $x \; E_0 \; y$ if and only if $x(n) = y(n)$ for almost all n. Next, $\cong_{\mathrm{HC}}$ denotes the isomorphism relation restricted to the set of codes for hereditarily countable sets. Finally, E_{set} denotes the relation defined by $x \; E_{\mathrm{set}} \; y$ if x and y, thought of as codes for countable sequences of reals, enumerate the same set.

Theorem 10.

(1) *E_0 does not infinite time computably reduce to $=$.*
(2) *E_{set} does not infinite time computably reduce to E_0.*
(3) *$\cong_{\mathrm{HC}}$ and E_{set} do not infinite time computably reduce to $\cong_{\mathrm{WO}}$.*

Without strong set-theoretic hypotheses, such results cannot be obtained for reduction functions which are much more general than the absolutely $\undertilde{\Delta}^1_2$ functions. For instance, the infinite time semi-computable reduction functions are still well inside the class $\undertilde{\Delta}^1_2$, but if we were to allow reduction functions in this class, then all of the equivalence relations in Theorem 10 would be reducible to the equality relation.

Theorem 11. *If $V = L$, then every infinite time computable equivalence relation on $\mathbb{R}$ is reducible to the equality relation by an infinite time semi-computable function.*

The proof of Theorem 11 uses the same ideas as in the proof of Theorem 6, and as in that argument, the reduction functions are not selectors for the relation. On the other hand, under suitable determinacy hypotheses, every infinite time semi-computable function is Lebesgue measurable. In this situation, infinite time semi-computable reducibility again resembles the more concrete reducibility notions.

We have seen that by expanding the class of reduction functions available, we are sometimes able to bring a wider class of equivalence relations under consideration. A major example of this is the following generalization of the class of countable Borel equivalence relations. Here, a Borel equivalence relation is said to be *countable* iff every equivalence class is countable. The countable relations have become one of the most important collections studied in the classical theory, since many natural relations lie at this level and some basic progress has been made in uncovering their structure under $\leq_B$. For instance, by a classical result of Silver, the equality relation $=$ is the $\leq_B$-least countable Borel equivalence relation. Moreover, by a deep result of Kechris-Harrington-Louveau, E_0 is the $\leq_B$-least Borel equivalence relation which is not reducible to $=$. Thirdly, we have that there is a $\leq_B$-greatest countable Borel equivalence relation, denoted E_∞. The remaining countable Borel equivalence relations lie in the interval (E_0, E_∞), and a result of Adams-Kechris implies that there are continuum many distinct relations up to Borel bireducibility.

This last result holds also in the context of $\leq_c$ and $\leq_e$ reducibility, since the arguments that Adams and Kechris use to establish non-reducibility are measure-theoretic. We presently define a class of infinite time computable relations which we propose is the correct analogue of the countable Borel equivalence relations, and investigate the corresponding generalizations of the remaining results. The idea comes from the classical proof of the maximality of E_∞, which hinges on the following characterization of the countable Borel equivalence relations. Namely, E is a countable Borel equivalence relation if and only if it admits a *Borel enumeration*, that is, a Borel function f such that $f(x)$ codes an enumeration of $[x]_E$, for all x. (This characterization is an immediate consequence of the Lusin-Novikov theorem from descriptive set theory.) Generalizing this, we say that the equivalence relation E is (infinite time) *enumerable* if there exists an infinite time computable function f such that $f(x)$ codes an enumeration of $[x]_E$, for all x. The *eventually enumerable* equivalence relations are defined analogously. This is a worthwhile generalization; for instance the relation defined by $x \equiv_{\mathsf{hyp}} y$ iff x and y are hyperarithmetic in one another is enumerable but not Borel.

Since we have said that the maximality of E_∞ depends on the above characterization of the countable Borel equivalence relations, and since we have defined the enumerable and eventually enumerable equivalence relations in the analogous way, the proof of maximality of E_∞ in the Borel context yields the same in our context.

Theorem 12. *E_∞ is $\leq_c$-greatest among the enumerable relations, and $\leq_e$-greatest among the eventually enumerable relations.*

Perhaps surprisingly, one can establish the minimality of $=$ as well.

Theorem 13. *$=$ is reducible to every eventually enumerable equivalence relation by a continuous function.*

This result is an immediate consequence of the fact (due originally to Welch) that there exists a perfect set of $\equiv_{e\infty}$-classes. (Here, $\equiv_{e\infty}$ denotes the *eventual degree* relation, which is defined analogously to $\equiv_\infty$.) The idea of Welch's proof is to use the theory of forcing over L_Σ to obtain a perfect set of mutually generic Cohen reals, and then argue that this set does the job. To see that Theorem 13 follows, observe that every eventually enumerable relation E is contained (as a set of pairs) in the relation $\equiv_{e\infty}$. Hence there exists a perfect set of E-classes, and it follows that there is a continuous reduction from $=$ to E.

Finally, we have been unable to establish the minimality of E_0 over the equality relation, and we leave this as a question. It is hoped that methods similar to the proof of Theorem 13 will provide an answer.

Question 14. *Is it true of every enumerable equivalence relation E that either E is reducible to $=$ or else E_0 is reducible to E?*

REFERENCES

[CFK$^+$10] Merlin Carl, Tim Fischbach, Peter Koepke, Russell Miller, Miriam Nasfi, and Gregor Weckbecker. The basic theory of infinite time register machines. *Arch. Math. Logic*, 49(2):249–273, 2010.

[CH11] Sam Coskey and Joel David Hamkins. Infinite time computable equivalence relations. *Notre Dame Journal of Formal Logic*, 2011. to appear.

[DHS05] Vinay Deolalikar, Joel David Hamkins, and Ralf-Dieter Schindler. $\mathrm{P} \neq \mathrm{NP} \cap \mathrm{co} - \mathrm{NP}$ for infinite time turing machines. *Journal of Logic and Computation*, 15(5):577–592, October 2005.

[FS89] Harvey Friedman and Lee Stanley. A Borel reducibility theory for classes of countable structures. *J. Symbolic Logic*, 54(3):894–914, 1989.

[Ham02] Joel David Hamkins. Infinite time turing machines. *Minds and Machines*, 12(4):521–539, 2002. (special issue devoted to hypercomputation).

[Ham04] Joel David Hamkins. Supertask computation. In Boris Piwinger Benedikt Löwe and Thoralf Räsch, editors, *Classical and New Paradigms of Computation and their Complexity Hierarchies*, volume 23 of *Trends in Logic*, pages 141–158. Kluwer Academic Publishers, 2004. Papers of the conference "Foundations of the Formal Sciences III" held in Vienna, September 21-24, 2001.

[Ham05] Joel David Hamkins. Infinitary computability with infinite time Turing machines. In Barry S. Cooper and Benedikt Löwe, editors, *New Computational Paradigms*, volume 3526 of *LNCS*, Amsterdam, June 8-12 2005. CiE, Springer-Verlag.

[Ham07] Joel David Hamkins. A survey of infinite time Turing machines. In Jérôme Durand-Lose and Maurice Margenstern, editors, *Machines, Computations, and Universality - 5th International Conference MCU 2007*, volume 4664 of *Lecture Notes in Computer Science*, Orleans, France, 2007.

[Hjo00] Greg Hjorth. *Classification and orbit equivalence relations*, volume 75 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 2000.

[HK96] Greg Hjorth and Alexander S. Kechris. Borel equivalence relations and classifications of countable models. *Ann. Pure Appl. Logic*, 82(3):221–272, 1996.

[HL00] Joel David Hamkins and Andy Lewis. Infinite time Turing machines. *J. Symbolic Logic*, 65(2):567–604, 2000.

[HL02] Joel David Hamkins and Andy Lewis. Post's problem for supertasks has both positive and negative solutions. *Archive for Mathematical Logic*, 41(6):507–523, 2002.

[HLM07] Joel David Hamkins, David Linetsky, and Russell Miller. The complexity of quickly decidable ORM-decidable sets. In Barry Cooper, Benedikt Löwe, and Andrea Sorbi, editors, *Computation and Logic in the Real World - Third Conference of Computability in Europe CiE 2007*, volume 4497 of *Proceedings, Lecture Notes in Computer Science*, Siena, Italy, 2007.

[HM07] Joel David Hamkins and Russell Miller. Post's problem for ordinal register machines. In Barry Cooper, Benedikt Löwe, and Andrea Sorbi, editors, *Computation and Logic in the Real World - Third Conference of Computability in Europe CiE 2007*, volume 4497 of *Proceedings, Lecture Notes in Computer Science*, Siena, Italy, 2007.

[HM09] Joel David Hamkins and Russell Miller. Post's problem for ordinal register machines: an explicit approach. *Annals of Pure and Applied Logic*, 160(3):302–309, 2009.

[HMSW07] J. D. Hamkins, R. Miller, D. Seabold, and S. Warner. Infinite time computable model theory. In S.B. Cooper, Benedikt Löwe, and Andrea Sorbi, editors, *New Computational Paradigms: Changing Conceptions of What is Computable*, pages 521–557. Springer, 2007.

[HS01] Joel David Hamkins and Daniel Seabold. Infinite time Turing machines with only one tape. *Mathematical Logic Quarterly*, 47(2):271–287, 2001.

[HW03] Joel David Hamkins and Philip Welch. $P^f \neq NP^f$ for almost all f. *Mathematical Logic Quarterly*, 49(5):536–540, 2003.

[KK06] Peter Koepke and Martin Koerwien. Ordinal computations. *Math. Structures Comput. Sci.*, 16(5):867–884, 2006.

[Koe05] Peter Koepke. Turing computations on ordinals. *Bulletin of Symbolic Logic*, 11(3):377–397, September 2005.

[KS06] Peter Koepke and Ryan Siders. Register computations on ordinals. *submitted to: Archive for Mathematical Logic*, 2006.

[KS09] Peter Koepke and Benjamin Seyfferth. Ordinal machines and admissible recursion theory. *Ann. Pure Appl. Logic*, 160(3):310–318, 2009.

[LÖ1] Benedikt Löwe. Revision sequences and computers with an infinite amount of time. *Logic Comput.*, 11(1):25–40, 2001.

[LM04] Giacomo Lenzi and Erich Monteleone. On fixpoint arithmetic and infinite time turing machines. *Information Processing Letters*, 91(3):121–128, 2004.

[Sch03] Ralf-Dieter Schindler. P $\neq$ NP for infinite time Turing machines. *Monatshefte für Mathematik*, 139(4):335–340, 2003.

[ST11] Scott Schneider and Simon Thomas. Countable Borel equivalence relations. In *Appalachian Set Theory 2006–2010*, 2011.

[Wel] Philip Welch. On a question of Deolalikar, Hamkins and Schindler. available on the author's web page at http://www2.maths.bris.ac.uk/~mapdw/dhs.ps.

[Wel99] Philip Welch. Friedman's trick: Minimality arguments in the infinite time Turing degrees. *in "Sets and Proofs", Proceedings ASL Logic Colloquium*, 258:425–436, 1999.

[Wel00a] Philip Welch. Eventually infinite time Turing machine degrees: Infinite time decidable reals. *Journal of Symbolic Logic*, 65(3):1193–1203, 2000.

[Wel00b] Philip Welch. The lengths of infinite time Turing machine computations. *Bulletin of the London Mathematical Society*, 32(2):129–136, 2000.

[Wel05] Philip Welch. The transfinite action of 1 tape Turing machines. In Barry S. Cooper and Benedikt Löwe, editors, *New Computational Paradigms*, volume 3526 of *LNCS*, Amsterdam, June 8-12 2005. CiE, Springer-Verlag.

THE FIELDS INSTITUTE, 222 COLLEGE STREET, TORONTO ONTARIO M5T 3J1 & THE YORK UNIVERSITY DEPARTMENT OF MATHEMATICS AND STATISTICS, N520 ROSS, 4700 KEELE STREET, TORONTO, ONTARIO M3J 1P3

E-mail address: scoskey@nylogic.org, http://math.rutgers.edu/∼scoskey

THE GRADUATE CENTER OF THE CITY UNIVERSITY OF NEW YORK, MATHEMATICS PROGRAM, 365 FIFTH AVENUE, NEW YORK, NY 10016 & COLLEGE OF STATEN ISLAND OF CUNY, MATHEMATICS, 2800 VICTORY BOULEVARD, STATEN ISLAND, NY 10314

E-mail address: jhamkins@gc.cuny.edu, http://jdh.hamkins.org